

ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО
АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2
2021-2022

- други домаћи задатак -

Опште напомене:

1. Пре одбране сви студенти раде тест знања који се ради на рачунару коришћењем система Moodle (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да креирају налог и пријаве се на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити **прихваћена и важећа** само уколико се студент региструје путем свог налога електронске поште на серверу **mail.student.etf.bg.ac.rs**.
2. Домаћи задатак 2 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику C++.
3. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Унос података треба омогућити било путем читања са стандардног улаза, било путем читања из датотеке.
5. Решења треба да буду отпорна на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
6. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Примена рекурзије се неће признати као решење проблема које може освојити максималан број поена.**
7. За све недовољно јасне захтеве у задатку, студенти треба да усвоје разумну претпоставку у вези реализације програма. Приликом одбране, демонстраторе треба обавестити која претпоставка је усвојена (или које претпоставке су усвојене) и која су ограничења програма (на пример, максимална димензија низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
8. Одбрана другог домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета. Пријава за одбрану биће омогућена преко Moodle система. Детаљније информације биће објављене на предметном сајту.
9. Предаја домаћих ће бити омогућена преко Moodle система. Детаљније информације ће бити благовремено објављене.
10. За решавање задатака који имају више комбинација користити следеће формуле.
(**R** – редни број индекса, **G** – последње две цифре године уписа):
$$i = (R + G) \bmod 2$$
11. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака, као и да пријаве теже случајеве повреде Правилника о дисциплинској одговорности студената Универзитета у Београду Дисциплинској комисији Факултета.

Задатак 1 – Б* стабло [100 поена]

Написати на језику C++ класу која имплементира Б* стабло у које се умећу речи (стрингови до 256 карактера) као кључеви. Поређење стрингова вршити лексикографски. Ред стабла је у опсегу између 3 и 10, укључујући те границе. Реализовати следеће операције за рад са стаблом:

1. [15 поена] Стварање објекта стабла **задатог реда** и брисање објекта стабла
2. [15 поена] Проналажење кључа у стаблу (враћа вредност `true` ако кључ постоји, односно `false` у супротном)
3. [15 поена] Исписивање садржаја стабла у излазни ток (оператор `<<`), по нивоима, тако да су поједини чворови уочљиви (међусобно размакнути)
4. [15 поена] Уметање кључа у стабло (враћа вредност `false` ако кључ већ постоји, односно `true` у случају успешног уметања)
5. [20 поена] Брисање кључа из стабла (враћа вредност `false` ако кључ не постоји, односно `true` у случају успешног брисања)
6. [15 поена] У зависности од редног броја проблема *i* потребно је додатно имплементирати следећу функционалност:

0. Пронаћи к-ти најмањи лексикографски кључ у стаблу

1. За прослеђени кључ *X*, исписати колико постоји кључева у стаблу који су лексикографски мањи од прослеђеног *X*

[5 поена] Корисник са програмом интерагује путем једноставног менија. Програм треба да испише садржај менија, а затим да чека да корисник изабере (унесе путем тастатуре) редни број неке од понуђених ставки, након чега, пре извршења, од корисника очекује да по потреби унесе додатне параметре. Поступак се понавља све док корисник у менију не изабере опцију за прекид програма.

Напомене

По потреби реализовати и додатне методе, где је то примерено. Написати главне програме који кориснику омогућавају рад са задатим стаблом, путем једноставног интерактивног менија.

За тестирање програма се могу користити датотеке са кључевима које се налазе у оквиру посебне архиве. Датотеке садрже примере кључева генерисаних у различитом опсегу са униформном и нормалном дистрибуцијом.

Рад са датотекама у језику C++ захтева увожење заглавља `fstream` (именски простор `std`). За читање података користи се класа `ifstream`. Након отварања датотеке, читање се врши на исти начин као и са стандардног улаза. Кратак преглед најбитнијих метода и пријатељских функција ове класе је дат у наставку.

<pre>void open(const char *_Filename, ios_base::openmode _Mode = ios_base::in, int _Prot = (int)ios_base::_Openprot);</pre>	Отвара датотеку задатог имена за читање. ifstream dat; dat.open("datoteka.txt");
<pre>void close();</pre>	Затвара датотеку.
<pre>bool is_open();</pre>	Утврђује да ли је датотека отворена.
<pre>operator>></pre>	Преклопљен оператор за просте типове података.
<pre>ifstream dat; dat.open("datoteka.dat"); if(! dat.is_open()) greska(); char niz[20]; dat >> niz; dat.close();</pre>	Пример отварања датотеке, провере да ли је отварање успешно, читање једног знаковоног низа из датотеке и затварања датотеке.